

Санкт-Петербургский государственный университет

Кафедра технологии программирования

Столов Игорь Михайлович

Дипломная работа

**Реализация алгоритма поиска ассоциативных правил в формировании
корзины покупателя**

Научный руководитель,

ст. преподаватель

Стученков А.Б.

Санкт-Петербург

2016

Содержание

Содержание	2
Введение	3
Постановка задачи	5
Глава 1. Обзор задач, решаемых методами Data Mining	6
1.1. Решаемые задачи	6
1.2. Преимущества	7
Глава 2. Применение алгоритмов Data Mining в электронной коммерции	8
2.1. Описание и виды электронной коммерции	8
2.2. Применение Data Mining в электронной коммерции	9
Глава 3. Реализация алгоритма ассоциативных правил	12
3.1. Задача анализа корзины покупателя	12
3.1.1. Описание задачи и алгоритма решения	12
3.1.2. Пример поиска ассоциативных правил	14
3.2. Алгоритм Apriori	16
3.3. Программная реализация	21
3.3.1. Выбор средств разработки	21
3.3.2. Программная модель	22
3.3.3. Пример интеграции в логику работы сайта	22
3.3.4. Модуль для интернет-магазина	23
Выводы	30
Заключение	31
Список литературы	32
Приложение 1. Исходный код php-класса Arules	33
Приложение 2. Структура таблиц PHP-модуля	38

Введение

В наше время процесс сбора данных стал неотъемлемой частью практически всех областей человеческой деятельности. Бизнес, торговля, обучение, медицина – компании в этих и многих других сферах целенаправленно или попутно регистрируют огромное количество самой разнообразной информации – данные о финансах, клиентах, покупках, заказах, перевозках и т.д.

При наличии большого количества данных зачастую возникает проблема их обработки, а также появляется вопрос: возможно ли извлечь из собранных данных новую, нетривиальную и полезную информацию, которую можно было бы использовать в деятельности компании?

С задачей обработки позволяют справиться различные технологии анализа данных, например, OLAP, предназначенной для быстрой обработки сложных запросов к базе данных и служащей для подготовки бизнес-отчётов, например, по продажам и маркетингу.

А на последний вопрос дает ответ технология Data Mining (DM), представляющая собой “набор различных методов и алгоритмов для обнаружения в сырых данных ранее неизвестных, нетривиальных, практически полезных и доступных интерпретации знаний, необходимых для принятия решений в различных сферах человеческой деятельности”^[12].

На сегодняшний день есть много готовых решений для анализа данных средствами Data Mining, но почти все они распространяются на коммерческой основе за немалые деньги. При этом существует достаточно много небольших организаций, не готовых платить круглые суммы за мощные аналитические пакеты, но желающих использовать Data Mining в своей деятельности. К тому же, зачастую не требуется полный комплекс средств для анализа, а только один-два алгоритма.

Еще одной трудностью в использовании DM является необходимость наличия аналитика, умеющего работать со средствами анализа, знающего специфику настройки алгоритмов, способного должным образом подготовить данные. Очевидно, что не все организации способны держать в штате такого сотрудника.

Так же стоит отметить, что большое число возникающих организаций, являясь представителями малого бизнеса, для своей деятельности выбирают область электронной коммерции, минимизируя затраты на реализацию продукции.

Таким образом, до сих пор является актуальной задача разработки новых и реализации уже существующих алгоритмов анализа данных для нужд небольших организаций (действующих в области электронной коммерции), предоставляемых за небольшие деньги и обеспечивающих простое и удобное интегрирование в процесс работы компании.

Постановка задачи

Целью данной работы является теоретическое исследование области применения алгоритмов Data Mining в сфере электронной коммерции, а также практическая реализация алгоритма поиска ассоциативных правил с возможностью интеграции в существующую структуру данных и бизнес-процессов.

Для достижения поставленной цели в работе решаются следующие задачи:

- Изучение способов применения алгоритмов Data Mining в области электронной коммерции.
- Разработка PHP-класса, реализующего алгоритм Apriori (алгоритм поиска ассоциативных правил).
- Разработка интегрируемого программного модуля для интернет-магазина, выполняющего задачу анализа потребительской корзины покупателя на основе алгоритма ассоциативных правил.

Зачастую изменение функциональности, связанное с внедрением новых интеллектуальных алгоритмов, реализуется для уже существующих, активно работающих интернет-магазинов. Поэтому важной особенностью создаваемого решения должна быть гибкая интеграция в структуру существующего интернет-магазина.

Глава 1. Обзор задач, решаемых методами Data Mining

1.1. Решаемые задачи

Единого мнения относительно того, какие задачи следует относить к Data Mining, нет^{[3],[4],[21],[8]}. Ниже приводятся лишь основные из задач, решаемые при помощи DM.

- *Классификация*

Является достаточно простой и распространенной задачей DM. Решением задачи является обнаружение признаков, характеризующих классы набора данных, который мы исследуем. Следовательно, используя эти признаки, мы можем отнести новый исследуемый объект к определенному классу.

- *Кластеризация*

Являет собой не такую простую задачу, как классификация, в силу изначальной неопределенности классов объектов, то есть они изначально не заданы. Следствием решения задачи кластеризации считается разделение исследуемых объектов на определенные группы.

- *Ассоциация*

Решением задачи нахождения ассоциативных правил является поиск закономерностей в базах данных между взаимосвязанными событиями (например транзакциями).

- *Последовательность или последовательная ассоциация*

Последовательная ассоциация занимается поиском временных закономерностей между событиями (транзакциями).

Последовательность занимается нахождением закономерностей между связанными во времени транзакциями.

- *Прогнозирование (регрессия)*

Прогнозированием ведется оценка значений целевых численных показателей в будущем на базе особенностей полученных исторических данных.

1.2. Преимущества

В отличие от средств OLAP, в Data Mining переложены с человека на компьютер определение гипотез и нахождение нестандартных алгоритмов. При применении OLAP зачастую задаются вопросы наподобие: «Какое количество клиентов, не оплативших услугу?». При использовании Data Mining задаются вопросы наподобие «Какая категория клиентов не платит по счетам?». Отвечая на последний вопрос возможно нахождение нестандартного подхода в маркетинге и работе с заказчиками.

OLAP	Data Mining
Каковы средние показатели травматизма для курящих и некурящих?	Встречаются ли точные шаблоны в описаниях людей, подверженных повышенному травматизму?
Каковы средние размеры телефонных счетов существующих клиентов в сравнении со счетами бывших клиентов (отказавшихся от услуг телефонной компании)?	Имеются ли характерные портреты клиентов, которые, по всей вероятности, собираются отказаться от услуг телефонной компании?
Какова средняя величина ежедневных покупок по украденной и не украденной кредитной карточке?	Существуют ли стереотипные схемы покупок для случаев мошенничества с кредитными карточками?

Таблица 1.1. Формулировки задач при использовании методов OLAP и DM

Главное преимущество Data Mining состоит в нахождении нетривиальных шаблонов^[10], то есть в найденных шаблонах должны отражаться «скрытые знания», которые являют собой неожиданные и неочевидные регулярности в данных. Иными словами, средства DM отличаются от инструментов статистической обработки данных и средств OLAP тем, что вместо проверки заранее предполагаемых пользователями взаимозависимостей они на основании имеющихся данных способны находить такие взаимозависимости самостоятельно и строить гипотезы об их характере.

Глава 2. Применение алгоритмов Data Mining в электронной коммерции

2.1. Описание и виды электронной коммерции

Электронная коммерция является экономической сферой, включающей финансовые операции и бизнес процессы, которые осуществляются с помощью компьютерных сетей.

В настоящий момент электронная коммерция подразделяется на несколько общепринятых категорий, соответствующих целевым группам потребителей:

- **Схема B2B (бизнес-бизнес)** – схема, при которой между собой взаимодействуют два юридических лица (предприятия). Эта схема сегодня является очень перспективной и быстро развивающейся. Проведение всевозможных торговых операций значительно упрощается благодаря широким возможностям интернета. Пример работы по схеме B2B – создание и продажа шаблонов дизайнерской компанией для сайта другой компании с целью создания веб-сайта, который сам в свою очередь будет приносить деньги.
- **Схема B2C (бизнес-потребитель)** – смыслом этой схемы является прямое взаимодействие компании с клиентом. Клиенту такой способ торговли дает возможность упростить и ускорить процедуру покупки: нет необходимости идти в магазин, есть возможность ознакомиться с продуктом онлайн, прочитать характеристики и сделать заказ с возможностью доставки на дом. Предприниматель в свою очередь имеет возможность сэкономить на сотрудниках и размере офисного и складского помещений. Основным примером схемы B2C - интернет-магазины.
- **Схема C2C (потребитель-потребитель)** – при этой схеме подразумевается прямое взаимодействие между физическими лицами. Обычно коммерция по схеме C2C осуществляется на различных веб-сайтах (чаще всего - аукционы). Основное преимущество данной схемы состоит в более приемлемых ценах на товары.

Если говорить о видах электронной коммерции, то самым популярным в наше время являются **интернет-магазины**. Но, в то же время, приобретают популярность и другие разновидности.

Также весьма популярным видом электронной коммерции стал сервис **микрорплатежей**. Смысл такого подхода состоит во взимании небольшой суммы с клиента и

предоставление ему платного контента, будь то музыка, фильмы, игры, софт и так далее. Профит от такого бизнеса получается за счет количества посетителей таких сайтов.

Очень важным видом электронной коммерции является **реклама**. Для любого бизнесмена Интернет – не только торговая площадка, но и рекламная, открывающая новые возможности для бизнеса. Следовательно, правильная подача компании на просторах Интернета является важным моментом для бизнеса.

Также стоит отметить **интернет-банкинг**. Этот вид электронной коммерции теперь представляет не только вариант работы клиента с реальным банком через браузер, не выходя из дома, но и вариант, когда речь идет о полностью виртуальных финансовых институтах. Это стало реально после формирования компаний, предложивших полностью электронные кошельки с реальными балансами. Также наряду с кошельками стали развиваться сервисы обмена электронных валют.

Стоит упомянуть устоявшийся вид электронной коммерции, такой как **продажа информации**. Смысл таких сервисов состоит во взимании с клиента некоторой суммы денег за либо доступ к контенту сайта (платная регистрация), либо к информации, предоставляемой данным ресурсом. Такими сайтами являются сайты знакомств, базы данных, соцсети и так далее.

2.2 Применение Data Mining в электронной коммерции

Основным объектом электронной коммерции для применения DM являются интернет-магазины. Здесь возможно использование практически каждого алгоритма анализа данных для повышения эффективности работы и увеличения количества продаж.

К примеру, на основании данных о проданных товарах возможно определение сезонности товаров при помощи алгоритма прогнозирования. В результате анализа владелец ресурса может проводить различные акции и скидки, прогнозировать увеличения и спады объемов продаж.

Благодаря перекрестному прогнозированию, возможно предсказание продаж одного товара, если есть данные о продажах другого.

Из информации о произведенных в магазине заказах можно получить большое количество разнообразной полезной информации при помощи DM. К тому же, в интернет-магазинах такая информация зачастую собирается автоматически.

Так, при помощи алгоритмов прогнозирования и кластеризации можно определить зависимость продажи товаров от определенного атрибута. Например, определив зависимость продаж от региона, владелец интернет магазина может отображать необходимые товары в целевых городах, уменьшая количество лишнего товара и увеличивая вероятность продажи.

Используя алгоритм поиска взаимосвязей, можно определить группы связанных товаров, которые обычно покупаются вместе. Товары из одной группы можно размещать на одной странице, тем самым увеличивается количество перекрестных продаж.

Собранная информация о действиях клиента на ресурсе и посещенных им страницах впоследствии может использоваться в алгоритме кластеризации последовательностей. В результате анализа можно определить группы клиентов, для которых существует схожий маршрут передвижения по сайту, и отображать необходимые ссылки и товары для новых клиентов из этих групп.

Информация о клиентах бывает очень полезна и в других видах электронной коммерции. В первую очередь, это интернет-банкинг. Благодаря алгоритмам DataMining можно получить прогноз на действия нового клиента (к примеру, отдаст ли он кредит), а так же составить портрет типичных неплательщиков.

Разбиение клиентов на группы (к примеру, по возрасту), может помочь в определение связей между различными атрибутами и дать возможность сформировать индивидуальные подходы и общение с клиентами для увеличения количества успешных сделок.

Очень эффективным применение DM может быть и в области, напрямую не всегда относящейся к электронной коммерции, но тесно с ней связанной – это веб-голосование. В случае открытой системы учета голосов (не требующей регистрации на сайте), часто возникает много проблем с так называемой «накруткой» - искусственным увеличением числа голосов при помощи различных роботов. И способов борьбы с этой проблемой

довольно мало, так как нельзя всегда учесть все возможные варианты действия «накрутчиков».

Самым лучшим методом в борьбе с лишними голосами является анализ данных. Гораздо проще и эффективнее дать голосовать всем сколько угодно, но затем тщательно оценить полученные голоса. При ведении подробного учета информации о каждом поступившем голосе при помощи алгоритмов DM можно получить много полезных данных: динамику роста голосов, прогноз на дальнейшее поведение голосующих, возможные сценарии «накрутки голосов» и много другой полезной информации, на основе которой уже можно принять решения о добросовестности участников.

В данной работе основным объектом применения технологий Data Mining будет схема взаимодействия B2C, и, в частности, интернет магазины. А основным алгоритмом рассмотрения является алгоритм ассоциативных правил.

Глава 3. Реализация алгоритма ассоциативных правил

3.1. Задача анализа корзины покупателя

3.1.1. Описание задачи и алгоритма решения

Алгоритм нахождения ассоциативных правил взаимосвязан с термином анализа рыночной корзины (market basket analysis). Данный алгоритм был разработан для определения часто встречающихся паттернов (транзакций) для ритейлеров.

Постановка задачи может быть представлена как: допустим, существует БД из клиентских покупок. Каждая покупка – это транзакция, определяющая набор элементов (товаров), приобретенных клиентом за один раз.

Определение: «Пусть $I = \{i_1, i_2, i_3, \dots, i_n\}$ - множество товаров (элементы). Пусть D - множество транзакций, где каждая транзакция T – это набор элементов из I , $T \subseteq I$. Каждая транзакция представляет собой бинарный вектор, где $t[k]=1$, если i_k элемент присутствует в транзакции, иначе $t[k]=0$. Транзакция T содержит X , некоторый набор элементов из I , если $X \subset T$.»^[19]

Ассоциативное правило определяется как импликация $X \Rightarrow Y$, где $X \subset I$, $Y \subset I$ и $X \cap Y = \emptyset$. Поддержкой правила $X \Rightarrow Y$ называется величина support s , если $s\%$ транзакций из D , содержат $X \cup Y$,

$$\text{supp}(X \Rightarrow Y) = \text{supp}(X \cup Y)$$

Достоверность правила определяет то, какова вероятность того, что из X следует Y . Достоверностью правила $X \Rightarrow Y$ называется величина confidence c , если $c\%$ транзакций из D , содержащих X , также содержат Y ,

$$\text{conf}(X \Rightarrow Y) = \frac{\text{supp}(X \cup Y)}{\text{supp}(X)}$$

Приведем небольшой пример: 60% транзакций, в которых имеется велосипед, также имеется велосипедный шлем. В 7% всех транзакций есть оба элемента. 7% это поддержка, 60% - это достоверность правила, или "Велосипед" \Rightarrow "Шлем" с вероятностью 60%.

Таким образом, основной задачей алгоритма является нахождение следующих взаимосвязей: если в транзакции присутствует набор элементов X , то можно предположить, что набор элементов Y может также встретиться в этой транзакции.

Алгоритмы нахождения ассоциативных правил разработаны в первую очередь для определения всех возможных комбинаций (правил) $X \Rightarrow Y$ с поддержкой и достоверностью больше, чем заранее определенных пользователем порогов (thresholds), которые обозначают как минимальная поддержка и достоверность – minsupp и minconf .

Поиск ассоциативных правил состоит из двух подзадач:

- Поиск всех наборов элементов, удовлетворяющих minsupp threshold . Найденные наборы определяют как часто встречающиеся.
- Создание правил из найденных наборов элементов с достоверностью, которая удовлетворяет minconf threshold .

Некоторые реализации алгоритма ассоциативных правил вносят дополнительные характеристики, помимо поддержки и достоверности правила. Так, в службе Analysis Services MS SQL Server, алгоритм вычисляет для правила так называемую важность (importance). Находится эта величина следующим образом:

$$\text{Imp}(X \Rightarrow Y) = \log\left(\frac{p(X|Y)}{p(X|\text{not}Y)}\right)$$

Таким образом, в процессе вычисления уже задействованы вероятности, а значение важности правила представляет собой логарифм от отношения правдоподобия. $p(X|Y)$, $p(Y|\text{not}X)$ – условные вероятности.

Зачастую, важность правила представляет наибольший интерес при анализе, нежели достоверность. Важность отражает динамику изменения вероятности появления товара Y при выбранном товаре X . Если важность положительна, то товар Y скорее всего будет куплен, если куплен X . Если важность отрицательна, то товар Y вряд ли будет приобретен вместе с X . Нулевая важность говорит о том, что между товарами нет связи.

Помимо анализа рыночной корзины анализ ассоциативных правил применим к другим сферам, таким как биоинформатика, медицинские исследования, Web майнинг и анализ научных данных. Например, при исследовании данных о нашей планете,

ассоциативные шаблоны могут раскрывать интересные взаимосвязи между океаном, землей и атмосферными процессами. Такая информация может помочь ученым лучше понять, как разные элементы земли взаимодействуют и влияют друг на друга.

3.1.2. Пример поиска ассоциативных правил

В качестве примера работы алгоритма приведены результаты анализа данных компании, занимающейся продажей запчастей и аксессуаров к мобильным гаджетам. Основную деятельность компания ведет в области электронной коммерции, продавая продукцию через интернет-магазин.

Как было сказано ранее, алгоритм ассоциативных правил работает в два этапа. На первом этапе формируются часто встречающиеся наборы элементов, поддержка которых удовлетворяет минимальному порогу.

Поддержка	Товары
54	Корпус для iPhone 5S (Черный)
45	Корпус для iPhone 5S
43	Аккумулятор для iPhone 5 Li1450 + инструменты для вскрытия
32	Держатель SIM-карты iPhone 5
30	Комплект зарядного устройства в автомобиль с кабелем Apple 8pin USB 2,1A
27	Кабель с выходом USB для iPhone 8pin
26	Зарядное устройство в автомобиль с кабелем Apple 8 pin 2,1A
25	Кнопка Home для iPhone 4G
24	АКБ Samsung S6
23	АКБ Samsung NOTE 2N
22	Bluetoothmini клавиатура ST-BRK3100BT 49 клавиш черная
20	Беспроводные наушники MRH-8001+ складные черные: microSD, радио

Таблица 3.1. Часто встречающиеся наборы элементов, сформированные алгоритмом ассоциативных правил

Второй этап алгоритма – формирование правил из полученных наборов.

Достоверность	Важность	Правило
1	2,86707	Корпус для iPhone 5S (Черный)->Аккумулятор для iPhone 5 Li1450 + инструменты для вскрытия
1	1,880908	Корпус для iPhone 5, Аккумулятор для iPhone 5 Li1450 + инструменты для вскрытия->Держатель SIM-карты iPhone 5
0,956	1,631395	Права на использование Windows7Pro, Работы по монтажу, подключению и настройке обор. - >DiskKitWindows7ProfessionalRussian
0,952	1,825875	Сканер штрих-кода Cipher 1000-RS , Работы по монтажу, подключению и настройке обор. -> Блок питания 5V 0,35A
0,909	1,669007	Принтер штрих-кода Eltron 2824 LP , Дисплей покупателя DSP 840B (белый) -> Компьютер PMK:Стандарт (гар.3 года) (поз.1)

Таблица 3.2. Правила, сформированные алгоритмом ассоциативных правил

Как видно из полученных результатов, алгоритм выявил правила с единичной вероятностью, связанные с корпусом телефона и аккумуляторной батареей. Эти правила интуитивно понятны.

Большинство других правил связано непосредственно с продукцией и дает компании важную информацию, применимую в ее дальнейшей деятельности.

3.2. Алгоритм Apriori

В настоящий момент размеры БД покупательских транзакций, например супермаркетов, могут быть очень большими (и они продолжают расти), поэтому требуются ресурсоемкие масштабируемые алгоритмы для определения ассоциативных правил за небольшой промежуток времени. Алгоритм Apriori^[21] был одним из первых алгоритмов, предложенных для решения этой задачи.

Для использования алгоритма надо привести данные к нормализованному бинарному виду и поменять у данных структуру.

Обычный вид БД:

Номер транзакции	Наименование элемента	Количество
1001	A	2
1001	D	3
1001	E	1
1002	A	2
1002	F	1
1003	B	2
1003	A	2
1003	C	2
...

Таблица 3.3. Стандартный вид таблицы транзакций (заказов) магазина

Необходимо привести данные к нормализованному виду:

TID	A	B	C	D	E	F	G	H	I	K	...
1001	1	0	0	1	1	0	0	0	0	0	...
1002	1	0	0	0	0	1	0	0	0	0	...
1003	1	1	1	0	0	0	0	0	1	0	...

Таблица 3.4. Нормализованный вид таблицы транзакций (заказов) магазина

Элементы в столбцах отражают элементы, содержащиеся во множестве транзакций D. Каждая строка принадлежит транзакции, где в каждом столбце стоит 1, если элемент есть в транзакции, и 0 если нет.

Исходный вид таблицы стандартного вида транзакций может и отличаться от того, что приведен в таблице. Суть в том, чтобы преобразовать данные в нормализованный вид, чтобы алгоритм был применим. Также, все элементы должны быть упорядочены в алфавитном порядке (lexicographic order).

Как было сказано ранее, алгоритм Apriori работает в 2 этапа:

- Нахождение всех часто встречающихся наборов элементов;
- Извлечение правил.

Число элементов в наборе обозначается размером набора, а k элементный набор – это набор, состоящий из k элементов.

Нахождение всех часто встречающихся наборов может быть очень затратной по времени и ресурсоемкой задачей. Самое простое решение – брутфорс всех возможных вариантов. Этот подход задействует 2^n вычислений (n – количество элементов). Для уменьшения количества операций нужно воспользоваться **свойством антимонотонности** алгоритма Apriori: поддержка любого набора элементов меньше либо равна минимальной поддержки любого из его подмножеств.

Приведем пример. Support набора из 3-х элементов {Пиво, Чипсы, Кефир} будет всегда меньше или равен поддержке 2-х элементных наборов {Пиво, Чипсы}, {Пиво, Кефир}, {Чипсы, Кефир}.

Это свойство служит для снижения размерности пространства поиска. Иначе перед нами стояла бы невыполнимая задача поиска многоэлементных наборов в связи с экспоненциальным ростом вычислений.

Свойство анти-монотонности описывается также другим способом: любой k -элементный набор будет часто встречающимся, когда все его $(k-1)$ -элементные подмножества также часто встречаются.

Представим допустимые наборы элементов $I = \{A, B, C, D\}$ в виде 3-х уровневой решетки, где на верхнем уровне пустое множество. На первом уровне в нашей решетке размещаются 1-элементные наборы, на втором – 2-х элементные и т.д. В общем случае, на k -ом уровне размещаются k элементные наборы, связанные со своими $(k-1)$ элементными подмножествами.

Проиллюстрируем нашу 3-х уровневую решетку. Допустим, что у набора из элементов $\{A, B\}$ имеется поддержка ниже заданного threshold и, следовательно, не является часто встречающимся. Используя свойство антимонотонности, можно утверждать, что супермножества от $\{A, B\}$ также не будут часто встречающимися. Все эти супермножества можно убрать из нашей решетки (выделено синим фоном). С таким подходом можно значительно сузить наш поиск.

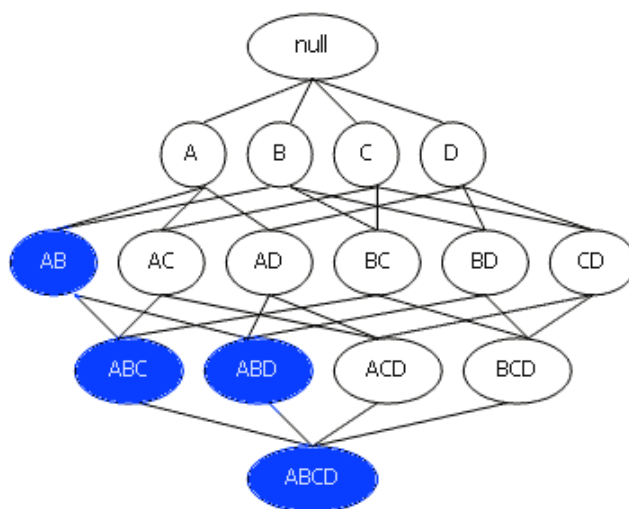


Рисунок 3.5. Иллюстрация принципа анти-монотонности

Рассмотрим работу алгоритма. Сначала подсчитываются все одно-элементные наборы и находятся все часто встречающиеся. Для этого алгоритм делает один полный проход по всей базе и считает поддержку каждого элемента.

Следующие шаги будут состоять из двух частей: генерации потенциально часто встречающихся наборов элементов (кандидатов) и подсчета поддержки для кандидатов.

Алгоритм можно записать в виде следующего псевдо-кода:

1. $F_1 = \{\text{часто встречающиеся 1-элементные наборы}\}$
2. для ($k=2$; $F_{k-1} \neq \emptyset$; $k++$) {
3. $C_k = \text{Apriorigen}(F_{k-1})$ // генерация кандидатов
4. для всех транзакций $t \in T$ {
5. $C_t = \text{subset}(C_k, t)$ // удаление избыточных правил
6. для всех кандидатов $c \in C_t$
7. $c.\text{count}++$
8. }
9. $F_k = \{c \in C_k \mid c.\text{count} \geq \text{minsupport}\}$ // отбор кандидатов
10. }
11. Результат $\cup F_k$

Алгоритм не работает с базой данных для создания кандидатов. Для создания k -элементного набора нам необходимы только $(k-1)$ элементные часто встречающиеся наборы.

Исходный набор хранится в упорядоченном виде. Генерация кандидатов состоит из двух шагов.

- Объединение.

Каждый кандидат C_k будет формироваться путем расширения часто встречающегося набора размера $(k-1)$ добавлением элемента из другого $(k-1)$ -элементного набора.

- Удаление избыточных правил.

На основании свойства анти-монотонности, удаляются все наборы $c \in C_k$ если хотя бы одно из его $(k-1)$ подмножеств не является часто встречающимся.

Следующим шагом после создания кандидатов нам следует посчитать поддержку для каждого из них. Однако, может получиться большое количество кандидатов и их подсчет

будет очень ресурсоемким. В этом случае необходим эффективный способ подсчета кандидатов. Самое простое решение – сравнение кандидата с каждой транзакцией. Но это потребует много ресурсов. Более качественное решение заключается в хранении кандидатов в хэш-дереве, в котором внутренние ноды (nodes) содержат хэш таблицы со ссылками на дочерние элементы, а листья - на кандидатов.

Каждое формирование кандидатов требует построение хэш-деревя. В начале, дерево – это только корень, который есть также лист, не имеющий никаких кандидатов-наборов. При каждом формировании нового кандидата он ставится в корень дерева и так пока число кандидатов в корневом элементе листе не превысит некоторое предельное значение(threshold). В тот момент, когда число кандидатов превышает некоторое предельное значение, корень превращается в хэш-таблицу, т.е. он уже есть ни что иное как внутренний элемент, и для него генерируются дочерние элементы листья. И все примеры (наборы) распределяются по дочерним элементам в соответствии с хэш-значениями элементов, состоящих в наборе, и так далее. Происходит каждого нового кандидата на внутренних элементах до тех пор, пока кандидат не попадет в первый элемент лист. Там он будет находиться, пока число наборов снова не станет больше предельного значения.

В тот момент, когда образовано хэш дерево, можно посчитать поддержку для всех кандидатов. С этой целью требуется пропустить все транзакции через дерево и повысить значение счетчиков для кандидатов, элементы которых также состоят в транзакции, т.е. $C_k \cap T_i = C_k$. На первом уровне значение хэшфункции вычисляется для каждого элемента из транзакции. Потом, на втором уровне, значение хэшфункции вычисляется для вторых элементов и так далее. На k-ом уровне хэшируется k-ый элемент. И так мы продолжаем до достижения листового элемента. Если кандидат, содержащийся в листе, есть ни что иное как подмножество исследуемой транзакции, то счетчик поддержки кандидата повышается на 1.

После прохождения всех транзакций исходного набора данных через дерево, осуществляется сверка значения поддержки кандидатов на превышение минимального предельного значения (threshold). Кандидаты, удовлетворяющие этому условию, выявляются как часто встречающиеся. Помимо этого сохраняется сама поддержка набора, так как это необходимо при извлечении правил. Те же самые действия применимы при поиске (k+1) элементных наборов и так далее.

После нахождения всех часто встречающихся наборов элементов можно переходить к генерации правил.

Генерация правил представляет собой менее ресурсоемкую задачу. Чтобы посчитать достоверность правила необходимо знать только поддержки набора и множества в условии правила. Допустим, для часто встречающегося набора {Хлеб, Пиво, Чипсы} извлечем правило $\{Хлеб, Пиво\} \Rightarrow \{Чипсы\}$ и посчитаем его достоверность. Так как мы знаем поддержку этого набора и {Хлеб, Пиво}, находящееся в условии, тоже часто встречающееся по свойству антимонотонности, следовательно, мы знаем его поддержку. В итоге мы можем посчитать достоверность правила без необходимости лишнего прохода по базе данных.

Для извлечения правила из часто встречающегося набора F необходимо и достаточно нахождение всех его непустых подмножеств. И для каждого подмножества s формируется правило $s \Rightarrow (F - s)$, если достоверность правила $\text{conf}(s \Rightarrow (F - s))$ не меньше порога minconf .

3.3. Программная реализация

3.3.1. Выбор средств разработки

Так как основным объектом для применения Data Mining в данной работе являются интернет магазины, то при выборе средств для разработки во внимание принимались основные платформы для их создания.

Для небольших организаций приоритетными являются бесплатные или недорогие коммерческие продукты для создания интернет магазинов. Большинство подобных продуктов разработаны на языке PHP. В качестве СУБД чаще всего используется MySQL ввиду широкой интегрированности с PHP. Так же некоторые продукты позволяют использовать другие некоммерческие (PostgreSQL, FireBird) и коммерческие (MSSQL, Oracle) СУБД.

В качестве средства для разработки в данной работе была выбрана связка PHP–MySQL, как наиболее распространённая для создания интернет магазинов. Так же, благодаря средствам PHP для работы с базами данных, возможно расширение разработки на другие СУБД.

3.3.2. Программная модель

В рамках данной работы алгоритм Apriori был реализован в виде PHP-класса.

Класс содержит четыре поля и четыре метода. В полях содержатся основные настройки для работы алгоритма: `transactionField` (поле транзакций), `productField` (поле продукции), `minSupport` (минимальная поддержка) и `minConf` (минимальная достоверность).

Среди методов класса основными являются четыре метода: `normalize`, `getOneElementSets`, `genSets`, `calculateRules`.

Метод `normalize` предназначен для подготовительного этапа работы алгоритма – нормализации таблицы заказов. Метод `getOneElementSets` формирует одноэлементные наборы товаров и оставляет часто встречающиеся на основе вычисленной поддержки набора.

Методы `genSets` и `calculateRules` реализуют в себе основную часть работы алгоритма Apriori.

Первый метод формирует часто встречающиеся наборы элементов на основе одноэлементных наборов и возвращает результат в виде ассоциативного массива.

Второй метод из сформированных наборов создает правила взаимосвязи между товарами и вычисляет достоверность для каждого правила. Результаты возвращаются в массиве, аналогично первому методу.

Исходный код класса Apriori с описанием методов приведен в приложении 1.

3.3.3. Пример интеграции в логику работы сайта

Класс Apriori легко интегрируется в логику работы интернет магазина и производит анализ данных о заказах. При ручной интеграции не играет роль СУБД сайта, так как методы алгоритма работают с уже выбранными данными.

Для проведения анализа алгоритму необходимо передать три массива данных – список заказов, список уникальных транзакций и список продукции. Они могут быть получены любым способом на усмотрение разработчика, использующего класс.

```
$orderList = getOrderList();
```

```
$tList = getTransactionList();
```

```
$pList = getProductsList();
```

Для начала анализа необходимо подключить модуль и создать новый объект класса Arules. В конструкторе необходимо задать параметры работы алгоритма – поле транзакций, поле продукции, минимальные поддержка и достоверность.

```
Include 'arules.php';
```

```
$arules = new Arules("transaction_id", "product", "0.1", "0.2");
```

Анализ данных проводится в четыре этапа. Первый этап – преобразование таблицы заказов в нормализованный вид. Для этого методу normalize передаются ранее сформированные данные.

```
$normOrders = $arules->normalize($orderList, $tList, $pList);
```

На втором этапе формируются часто встречающиеся одноэлементные наборы данных.

```
$oneElementSets = $arules->getOneElementSets($normOrders);
```

На третьем и четвертом этапах происходит формирование искомых данных – списка наборов и правил связанных товаров.

```
$sets = $arules->genSets($oneElementSets, $normOrders);
```

```
$rules = $arules->calculateRules($sets, $normOrders);
```

Переменные `set` и `rules` являются массивами с полученными результатами анализа. Эти данные можно сохранить в базу данных или записать в файл.

3.3.4. Модуль для интернет-магазина

Чтобы анализ данных был возможен без навыков в программировании для использования `php`-класса, в рамках данной работы был разработан специальный модуль для интернет магазина.

В отличие от ручного использования класса `Arules`, модуль имеет требования к СУБД интернет магазина, и для его использования необходимо наличие MySQL.

Установка модуля

Для установки модуля необходимо поместить его в директорию сайта и перейти по соответствующему адресу. При первом запуске модуль запрашивает необходимые данные

для подключения к базе данных, структуре таблиц, а так же позволяет сразу настроить параметры алгоритма.

Association Rules

Установка

Добро пожаловать в мастер установки модуля "Association Rules: поиск ассоциативных правил".
Заполните поля формы, чтобы мастер смог установить модуль на ваш сайт.

Подключение к БД

Хост:

Логин:

Пароль:

Имя базы данных:

Параметры базы данных

Таблица заказов:

Поле с кодом заказа:

Поле товара:

Настройки алгоритма ([Помощь](#))

Минимальная поддержка:

Минимальная достоверность:

Рисунок 3.6. Страница установки модуля

Данные о структуре таблицы заказов необходимы для работы алгоритма Apriori. Если пользователь не знаком с параметрами алгоритма, то он может воспользоваться помощью.

Во время установки модуля в базе данных создаются таблицы для хранения настроек алгоритма и модуля, сформированных наборов и правил и информации о проведенных анализах.

Подробная структура создаваемых таблиц описана в приложении 2.

Проведение анализа

После успешной установки модуля пользователь попадает на главную страницу. На ней отображается список проведенных анализов (если есть), и кнопка для проведения нового анализа. Так же пользователю доступно меню для навигации в модуле.

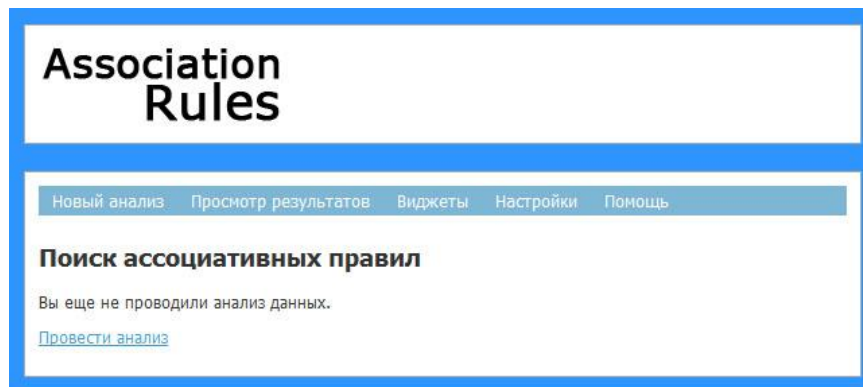


Рисунок 3.7. Главная страница модуля

Пункт **Новый анализ** ведет на страницу проведения анализа. В разделе **Просмотр результатов** пользователь может посмотреть полученные наборы и правила. Встраиваемые в интернет-магазин блоки для отображения связанных товаров создаются на странице **Виджеты**. В разделе **Настройки** можно изменить параметры работы алгоритма, а в разделе **Помощь** более подробно узнать о его работе.

В разделе «Новый анализ» пользователю предлагается провести анализ данных о заказах своего магазина. На основе структуры таблицы, описанной в настройках, модуль запустит алгоритм Apriori и сформирует наборы и правила для связанных товаров.

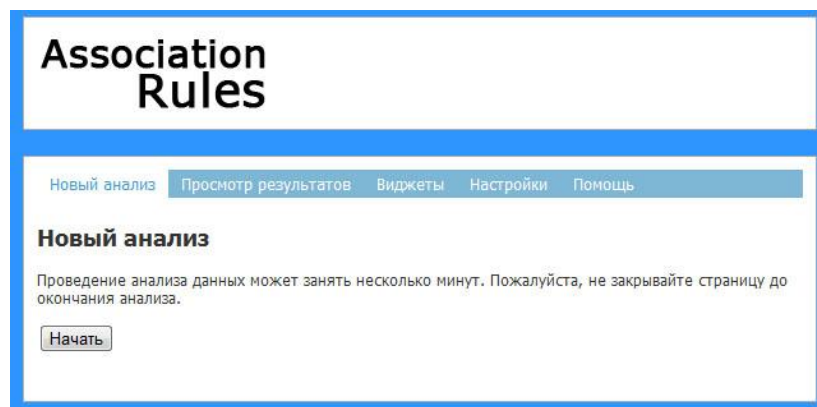


Рисунок 3.8. Страница проведения анализа

Просмотр результатов

После завершения анализа пользователь может перейти на страницу просмотра результатов. На вкладке «Наборы» отображаются все множества связанных элементов, которые выявил алгоритм. У каждого множества указана своя поддержка.

Association Rules		
<div> Новый анализ Просмотр результатов Виджеты Настройки Помощь </div>		
Результаты анализа		
Наборы Правила		
<div> Наборы Выгрузка в csv </div>		
#	Элементы	Поддержка
1	Денежный ящик DS-2045 330*360*90 24V, Сканер штрих-кода Spher 1000-RS	3
2	T/T этикетки (99*85/1000) п/п для Agroca, Термоэтикетки (100*125/500) TOP	3
3	Блок питания 5V 0,35A, Мышь Logitech Optical Mouse PS/2	2
4	Блок питания 5V 0,35A, Плата расширения PCI-2 COM(RS-232) NetMos	2
5	Блок питания 5V 0,35A, Фиск.регистратор ШТРИХ-ФР-К	2
6	Дисплей покупателя DSP 840B (белый), Плата расширения PCI-2 COM(RS-232) NetMos	2
7	Кабель интерфейсный USB (A->B), Компьютер Офис:Стандарт (гар.3 года)	2
8	Компьютер Офис:Стандарт (гар.3 года) (поз.2), Мышь Logitech Optical Mouse PS/2	2
9	Компьютер Офис:Стандарт (гар.3 года) (поз.2), Плата расширения PCI-2 COM(RS-232) NetMos	2
10	Компьютер Офис:Стандарт (гар.3 года) (поз.2), Сканер штрих-кода Spher 1000-RS	2
11	Мышь Logitech Optical Mouse PS/2, Плата расширения PCI-2 COM(RS-232) NetMos	2
12	Компьютер Офис:Стандарт (гар.3 года) (поз.2), Мышь Logitech Optical Mouse PS/2, Плата расширения PCI-2 COM(RS-232) NetMos	2

Рисунок 3.9. Страница со сформированными наборами

На вкладке «Правила» пользователь может посмотреть правила взаимосвязи между товарами, сформированные алгоритмами. Они отображаются в три колонки – левая часть правила, правая часть правила, достоверность.

Association Rules			
<div> Новый анализ Просмотр результатов Виджеты Настройки Помощь </div>			
Результаты анализа			
Наборы Правила			
<div> Правила Выгрузка в csv </div>			
#	Правило		Достоверность
1	Мышь Logitech Optical Mouse PS/2	=> Блок питания 5V 0,35A	0.67
2	Плата расширения PCI-2 COM(RS-232) NetMos	=> Блок питания 5V 0,35A	0.67
3	Плата расширения PCI-2 COM(RS-232) NetMos	=> Дисплей покупателя DSP 840B (белый)	0.67
4	Дисплей покупателя DSP 840B (белый)	=> Плата расширения PCI-2 COM(RS-232) NetMos	0.67
5	Мышь Logitech Optical Mouse PS/2	=> Компьютер Офис:Стандарт (гар.3 года) (поз.2)	0.67
6	Плата расширения PCI-2 COM(RS-232) NetMos	=> Компьютер Офис:Стандарт (гар.3 года) (поз.2)	0.67
7	Плата расширения PCI-2 COM(RS-232) NetMos	=> Мышь Logitech Optical Mouse PS/2	0.67
8	Мышь Logitech Optical Mouse PS/2	=> Плата расширения PCI-2 COM(RS-232) NetMos	0.67
9	Блок питания 5V 0,35A	=> Мышь Logitech Optical Mouse PS/2	0.50
10	Блок питания 5V 0,35A	=> Плата расширения PCI-2 COM(RS-232) NetMos	0.50

Рисунок 3.10. Страница со сформированными правилами

Полученные результаты пользователь может использовать по своему усмотрению двумя способами. Первый способ – выгрузка данных в csv файл. В случае наборов формат файла будет следующим:

элемент 1, элемент 2, ... ;поддержка

элемент 1, элемент 2, ... ;поддержка

Для правил формат файла выглядит так:

левый элемент 1, левый элемент 2,... ; правый элемент ; достоверность

Виджеты

Модуль предоставляет возможность использовать полученные результаты анализа прямо на страницах интернет магазина при помощи виджетов. На соответствующей странице модуля пользователь может выбрать два решения использования – на странице подробной информации о товаре и странице с корзиной покупателя.

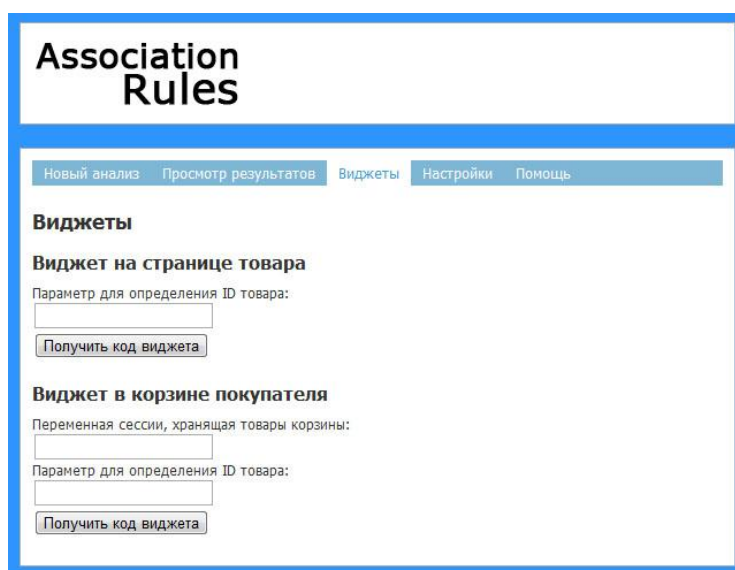


Рисунок 3.11. Страница виджетов

После заполнения основных настроек для виджета пользователю предоставляется код для вставки на соответствующую страницу.

Association Rules

Новый анализПросмотр результатовВиджетыНастройкиПомощь

Виджеты

Виджет на странице товара

Параметр для определения ID товара:

product

Получить код виджета

Вставьте этот код на страницу с подробной информацией о товаре:

```
$productParam = 'product';
include '/lib/widgets/productDetailPage.php';
```

Виджет в корзине покупателя

Переменная сессии, хранящая товары корзины:

cart

Параметр для определения ID товара:

product

Получить код виджета

Вставьте этот код на страницу с корзиной покупателя:

```
$cartParam = 'cart';
$productParam = 'product';
include '/lib/widgets/cartPage.php';
```

Рисунок 3.12. Страница виджетов с кодами для вставки

Виджет, установленный на странице с подробной информацией о товаре, производит поиск по таблице с наборами связанных товаров. На основе уникального ID товара, полученного из параметра, указанного в настройках, выбираются все товары, встречавшиеся вместе с искомым в транзакциях.

ОБРАТНАЯ СВЯЗЬ

Вход

Регистрация

Хочу!

Корпус для iPhone 5S (Черный)

ДЛЯ ТЕЛЕФОНОВ

Apple iPhone

Nokia

Samsung

LG

Sony

HTC

Huawei

ДЛЯ ПЛАНШЕТОВ

Apple iPad

ДЛЯ НОУТБУКОВ

Apple MacBook

3 500 руб Под заказ

ПОЛОЖИТЬ В КОРЗИНУ

КУПИТЬ В ОДИН КЛИК

ОЦЕНИТЕ ТОВАР:

Доставка

Самовывоз в Санкт-Петербурге

Доставка почтой по РФ

Артикул: 1002824

С этим покупают

Аккумулятор для iPhone 5 Li1450 + инструменты для вскрытия

990 руб

Держатель SIM-карты iPhone 5 черный

90 руб

Рисунок 3.13. Пример работы виджета на странице товара

Виджет, используемый в корзине покупателя, находит связанные товары из таблицы с правилами. Происходит поиск товаров, находящихся в корзине покупателя, в левой части правил, после чего пользователю предлагаются товары, находящиеся в правой части.

3 500 руб Под заказ

УЖЕ В КОРЗИНЕ

КУПИТЬ В ОДИН КЛИК

ОЦЕНИТЕ ТОВАР:

Доставка

Самовывоз в Санкт-Петербурге

Доставка почтой по РФ

Артикул: 1002823

Сопутствующие товары

Кнопка Home для iPhone 4G

290 руб

Металлический держатель СИМ-карты iPhone 4G

490 руб

Рисунок 3.14. Пример работы виджета в корзине покупателя

29

Выводы

В результате проделанной работы были решены следующие задачи:

- Изучены основные способы применения алгоритмов Data Mining в области электронной коммерции, выбраны виды и сферы, в которых наиболее применим анализ данных.
- Разработан PHP-класс, реализующий алгоритм ассоциативных правил Apriori, который производит поиск ассоциативных правил во множестве элементов.
- Разработан программный модуль, решающий задачу анализа корзины покупателя на основе нахождения ассоциативных правил алгоритмом Apriori. Модуль встраивается в интернет магазин, предоставляя интерфейс по настройке алгоритма и проведению анализа данных. Полученные правила могут использоваться на страницах магазина для подбора перекрестных товаров при помощи виджетов модуля или могут быть выгружены с сайта для дальнейшего использования.

Заключение

Решение, описанное в данной работе, применимо для большинства интернет-магазинов, реализованных на базе технологии PHP.

Разработанный программный модуль внедрен в практическую деятельность компании ООО «СИЭКСПО». Полученные в результате анализа данные позволили определить группы связанных товаров и увеличить эффективность торговли за счет перекрестных продаж.

Список литературы

1. Agrawal, Rakesh and Srikant, Ramakrishnan: Fast algorithms for mining association rules in large databases. Чили, 1994, с. 487-499
2. CharuC. Agrawal, Jiawei Han: Frequent Pattern Mining. Швейцария:Springer, 2014.
3. J.Han, M. Kamber. Data Mining: Concepts and Techniques. США, 2011.
4. А.А. Барсегян, М.С. Куприянов, В.В. Степаненко, И.И. Холод. Методы и модели анализа данных: OLAP и DataMining –СПб: БХВ-Петербург, 2008.
5. А. А. Барсегян, М. С. Куприянов, В. В. Степаненко, И. И. Холод. Технологии анализа данных. Data Mining, Visual Mining, Text Mining, OLAP – СПб: БХВ-Петербург, 2007 г.
6. Паклин Н.Б., Орешков В.И. Бизнес-аналитика: от данных к знаниям— СПб: Изд. Питер, 2009.
7. Antti Syvajarvi, Data Mining in Public and Private Sectors: Organizational and Government Applications (Premier Reference Source) - Information Science Reference, 2010
8. A.V. Senthil Kumar, Knowledge Discovery Practices and Emerging Applications of Data Mining: Trends and New Domains - IGI Global, 2010
9. Paolo Giudici, Applied Data Mining– Wiley, 2009
10. Чубукова И. А. DataMining: учебное пособие - М.: Интернет-университет информационных технологий: БИНОМ: Лаборатория знаний, 2006.
11. <http://www.intuit.ru/department/database/Data Mining/>
12. <http://ru.wikipedia.org>
13. http://www.sas.com/offices/europe/russia/software/enterprise_miner/index.html
14. <http://www.statsoft.com/products/statistica-data-miner/>
15. <http://www.spss.com/software/modeler/>
16. <http://www.oracle.com/>
17. <http://publib.boulder.ibm.com/>
18. <http://msdn.microsoft.com>
19. http://www.basegroup.ru/library/methodology/data_mining/

Приложение 1. Исходный код php-класса Arules

```
class Arules {

    public $transactionField;

    public $productField;

    public $minSupport;

    public $minConf;

    /* Конструктор*/

    public function Arules($tField, $pField, $minSupport, $minConf) {

        $this->transactionField = $tField;

        $this->productField = $pField;

        $this->minConf = $minConf;

        $this->minSupport = $minSupport;

    }

    /*

        * Приведение таблицы заказов к нормализованному виду.

        * Возвращаемый результат - массив - таблица, по строкам заказы,

        * по столбцам продукты, на пересечении 1, если товар

        * был в заказе, и 0 в противном случае

    */

    public function normalize($orderList, $transList, $productsList) {

        $result = array();

        foreach($transList as $item) {

            if(!isset($result[$item[$this->transactionField]])) {

                $result[$item[$this->transactionField]] = array();

                foreach ($productsList as $product) {
```

```

        if(in_array(array($this->transactionField=>$item[$this->transactionField], $this->productField=>$product[$this->productField]), $orderList))

            $result[$item[$this->transactionField]][$product[$this->productField]] = "1";

        else $result[$item[$this->transactionField]][$product[$this->productField]] = "0";
    }

    }

    }

    return $result;
}

/*
 * Формирует массив часто встречаемых одноэлементных наборов
 */

public function getOneElementSets($data) {

    $result = array();

    $index = 0;

    foreach ($data as $item) {

        foreach ($item as $product=>$value) {

            if(!isset($result[$index]["support"])) {

                $result[$index]["products"] = $product;

                $result[$index]["support"] = $value;

            }

            else $result[$index]["support"] += $value;

            $index++;

        }

        $index = 0;

    }
}

```

```

$finalResult = Array();

foreach ($result as $index=>$item) {

    $relSupport = $item["support"]/count($data);

    $row = $item;

    $row["rel_support"] = $relSupport;

    if($relSupport>= $this->minSupport)

$finalResult[] = $row;

}


    return $finalResult;

}


/*
*   Формирует часто встречающиеся множества
*/

public function genSets($oneElementSets, $data) {

    $sets = Array(0=>$oneElementSets);

    $setNum = 0;

    for($setCount=1; $setCount<3; $setCount++) {

        if(!empty ($sets[$setCount-1])) {

            foreach($sets[$setCount-1] as $index=>$set) {

                for($i=($index+1); $i<count($sets[0]); $i++) {

                    $newSet = Array();

```

```

        if (strpos($set["products"],
$sets[0][$i]["products"])===false) {

            $newSet["products"] = $set["products"].",
".$sets[0][$i]["products"];

            $newSet["support"] = $this-
>getSetSupport($newSet["products"], $data);

            $newSet["rel_support"] = $newSet["support"] /
count($data);

            if ($newSet["rel_support"] > $this->minSupport &&
!$this->isSetAlreadyExist($newSet["products"],
$sets[$setCount])) {

                $sets[$setCount][$setNum] = $newSet;

                $setNum++;

            }

        }

    }

}

unset($sets[0]);

return $sets;

}

/*
 * Формирует правила на основе полученных множеств
 */

public function calculateRules($fullSets, $data) {

    $ruleNum = 0;

    $setNumber = 0;

    $rules = Array();

    foreach($fullSets as $sets) {

```

```

foreach($sets as $set) {

    $ruleElems= explode(", ", $set["products"]);

    for($i=0; $i<count($ruleElems); $i++) {

        $elems = $ruleElems;

        $rightSide = $elems[$i];

        unset($elems[$i]);

        $leftSide = implode(", ", $elems);

        $support = $this->getSetSupport($leftSide,
        $data);

        if($support != 0) {

            $conf = $set["support"]/$support;

            if($conf> $this->minConf) {

                $rules[] = array("left"=>$leftSide,
                "right"=>$rightSide, "conf"=>$conf);

                $ruleNum++;

            }

        }

        $setNumber++;

    }

}

return $rules;

}

```

Приложение 2. Структура таблиц РНР-модуля

При установке модуля в базе данных интернет-магазина создаются несколько таблиц для хранения настроек модуля, сформированных наборах и правилах.

ar_config

- `param` varchar(100)
- `value` varchar(100)

Таблица хранит настройки работы алгоритма в виде пары «ключ - значение». Во время установки модуля в нее записываются данные о структуре таблиц с заказами (название, поле с заказом, поле с товаром) и настройки алгоритма Apriori (минимальные поддержка и достоверность).

ar_sessions

- `id` int(11)
- `sets` int(11)
- `rules` int(11)
- `date` timestamp

В данной таблице храниться информация о проведенных анализах данных. После завершения нового анализа в таблицу помещается количество полученных набор, правил и текущая дата и время.

ar_sets

- `id` int(11)
- `set` int(11)
- `product` varchar(200)
- `support` int(11)

Таблица предназначена для хранения полученных наборов. В процессе работы алгоритма каждый новый набор сохраняется в таблицу. Строка в таблице – продукт из набора, содержащая название товара, номер набора и его поддержку.

ar_rules

- `id` int(11)
- `set` int(11)
- `product` varchar(200)
- `type` varchar(2)
- `conf` float

В данной таблице сохраняются правила, получаемые в процессе работы алгоритма. Каждая строка – товар из правила, содержащая номер правила, название продукта, положение в правиле (левая или правая часть) и достоверность.